2000年度 大韓電子工學會 秋季綜合學術大會 論文集 第23卷 第2號 2000/11

타원곡선 암호 알고리즘에 기반한 digit-serial 승산기 설계

위 사 흔, *이 광 엽
서경대학교 컴퓨터과학과, *서경대학교 컴퓨터 공학과
전화 : 02-940-7240 / 핸드폰 : 016-369-3006

Design of digit-serial multiplier based on ECC(Elliptic Curve Cryptography) algorithm

Sa-Heun Wie, Kwang-Youb Lee
Dept. of Computer Science, seokyeong University
E-mail : gprix@home.skuniv.ac.kr

Abstract

소형화와 안전성에서 보다 더 진보된 ECC(Elliptic Curve Cryptography) 암호화 알고리즘의 하드웨어적 구현을 제안한다. Basis는 VLSI 구현에 적합한 standard basis이며 m=193 ECC 승산기 회로를 설계하였다. Bit-Parallel 구조를 바탕으로 Digit-Serial/Bit-Parallel 방법으로 구현하였다. 제안된 구조는 VHDL 및 SYNOPSYS로 검증되었다.

I. 서론

암호알고리즘은 키의 특성에 따라 크게 암·복호화 키가 같은 대칭키 암호알고리즘과 암·복호화 키가 서로 다른 공개키 암호알고리즘으로 나눌 수 있다. 그러나 암호 사용자가 늘어나고, 또한 다양한 암호서비스에 대한 요구가 제기되면서 대칭키 암호알고리즘에서 발생된 키 관리 문제와 인증 문제를 해결하기 위한 알고리즘의 필요성 대두되었다. 1976년 W. Diffie와 M. E. Hellman이 위의 두 문제를 해결한 "New Directions in Cryptography"에서 공개키 암호의 개념을 처음 소개하였다. 그 후 안전도 또는 실용적 측면에서 문제점을 유발한 많은 공개키 암호알고리즘이 제안되었고, 그중 1978년 소인수분해의 어려움에 기반을 둔 RSA가 소개되어 지금까지 넓게 사용되고 있다. 또한, 이산대수 문제에 기반을 둔 여러 가지 알고리즘들도 소개되었고, ECC 암호화 알고리즘을 위시한 여러 알고리즘 연구가 지속되어지고 있다.[1][2]

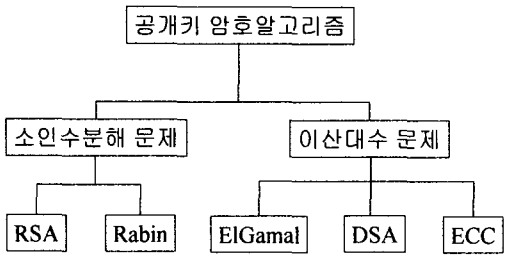

그림 1) 공개키 암호알고리즘 구성도

이 논문에서는 스마트 카드[3]와 같은 제한된 공간과 제한된 시간에서 보다 안전도가 높은 데이터 처리를 할 수 있는 ECC 암호화 알고리즘의 VLSI 구현을 제안하며 그중 승산기 회로를 설계 및 검증하며 나아가 제산기 및 그것들을 이용한 암 복호화 이론의 base를 제안한다.[6]

본 논문은 반도체 설계교육 센터(IDEC)의 지원장비를 활용하여 작성되었습니다.

II. 타원곡선

2,1 유한체와 타원곡선의 비교

유한체또는 Galois field 란 암호 이론이나 부호 이론에서 주로 사용되는 원소의 개수가 유한인 체(0에 의한 나눗셈을 제외하고는 4칙연산에 대해 닫혀있는 군)를 말한다. p를 소수라 하면 pro의 원소로 되는 유한체가 존재한다. 이것을 GF(p)로 쓰고 소체(prime field)라 부른다. 표현방식은 {0, 1, ···, p-1}이 된다. 이것으로부터 P(x)를 체 F위의 다항식이라고 하자. 이때 P(x)가 체 F 위에서 기약이면, P(x)를 모듈러로 하는 체 F 위의 다항식 환의 잉여류 환은 체가 된다는 것이 증명되어 있다. 또 GF(2)위에서 x^3+x+1은 기약이다. 이와 같은 다항식을 기약 다항식(irreducible polynomial)이라 한다 기약 다항식 P(x)의 차수를 n이라 하면, 같은 형태의 GF(p)에서 p^n개의 원소를 가지는 유한체를 만들 수 있다. 여기서 p=2로 놓으면 본 논문에서 논의되는 GF(2)가 되고 이것의 확장인 $GF(2^m)$이 되는 것이다. 이와 마찬가지로 유한체에서의 모듈러 p를 p(x)의 형태로 표현한다. $GF(2^m)$상에서의 소수 p(x)를 field generator로 사용한다.(본 논문에서는 $GF(2^{193})$에서 $p(x)=x^{193}+x^{15}+1$를 대상으로 한다)[7]

표 1) 유한체와 타원곡선의 비교

군	Z_p^*	$E(Z_p)$
군 원소 형태	정수 {1, 2, ... , p-1}	E의 점 (x, y)과 O
군 연산	모듈러 p에서의 곱셈	점의 더하기
표기	원소 : g, h 곱하기 : g*h 역원 : g^{-1} 나누기 : g/h 지수 : g^a	원소 : P, Q 더하기 : P + Q 역원 : - P 빼기 : P - Q 배수 : aP
이산 대수 문제	Z_p^*의 원소 g가 주어지고, $h = g^a$ mod p일 때, a를 찾는 문제	$E(Z_p)$의 원소 P가 주어지고, Q = aP일 때, a를 찾는 문제

표 1과 같이 타원곡선 암호시스템은 타원곡선 위의 점 P를 x번 더하는 계산이 주를 이룬다. 즉, $Q = xP$를 구하는 더하기 연산은 모듈러 곱셈을 통해 이루어진다. 주요 공개키 암호시스템은 효율적인 모듈러 곱셈에 의존하고 있으며, 타원곡선은 소수 p의 값이 다른 시스템보다 작기 때문에 그 효율성이 뛰어나다 할 수 있다. 즉, 타원곡선 암호시스템의 안전도는 타원곡선 이산대수문제에 의존하고 있으며, 그 효율성은 xP의 빠른 계산에 달려 있다. 위를 수행하는 알고리즘은 여러 가지로 변형될 수 있다.[6]

2.2 Polynomial Basis 구조

원소의 개수가 2^m개인 유한체 $GF(2^m)$상에서 다항식의 표현방법의 기본이 되는 basis는 NB(Normal Basis), PB(Polynomial Basis), 그리고 DB(dual Basis) 이렇게 크게 3가지로 나눌 수 있다. 표현방식을 보면 NB의 경우 $GF(2^m)$상에서 $\{\alpha, \alpha^2, \cdots, \alpha^{2^{m-1}}\}$이며 PB의 경우 $GF(2^m)$상에서 $\{1, \alpha, \cdots, \alpha^{m-1}\}$이다. DB는 $GF(2^m)$상에서 field generator P(x)의 근원인 α로 이루어진 PB $\{1, \alpha, \cdots, \alpha^{m-1}\}$를 바탕으로 $\{1, \alpha, \cdots, \alpha_{m-1}\}$로 표현된다.

위 형식에서 볼 때 NB에서 원소 a의 제곱근은 한번의 순회치환으로 얻어지므로 승산의 이점이 있다. 그러나 이 표현에 사용되는 함수 f에 의존한다는 단점이 있다. 또한 DB 표현도 게이트 수는 줄일 수 있으나 속도가 감소한다. 이 표현에 의한 하드웨어 구현시 DB로부터 PB로 변환하는 logic이 포함되어야 한다는 단점을 가진다. 이와 반대로 PB는 회로 구현시 구조가 간단하고 간단한 제어부를 요구한다. 또한 이 구조는 어떠한 field generator P(x)를 선택하더라도 동일한 구조를 사용할 수 있는 이점이 있다.

2.3 타원곡선 정의 및 타원곡선 암호시스템의 장점

타원곡선 암호시스템은 유한체의 곱셈군에 근거한 시스템으로써 다음의 장점을 가진다.

① 군(Group)을 제공할 수 있는 다양한 타원곡선을 활용할 수 있다. 즉, 다양한 암호시스템 설계가 용이하다.

② (초특이 타원곡선을 피하면)이 군에서의 subexponential time algorithms이 존재하지 않는다. 즉, 안전한 암호시스템을 설계하는 것이 용이하다.

③ 타원곡선 암호시스템은 존재하는 다른 공개키 스킴과 같은 안전도를 제공하는 데에 더 작은 키길이를 가지고 가능하다(예, RSA 1024비트 키와 ECC 160비

트 키를 갖는 암호시스템은 같은 안전도를 갖는다).

④ 타원곡선에서의 더하기 연산은 유한체에서의 연산을 포함하므로, H/W와 S/W로 구현하기가 용이하다. 더욱이 이 군에서의 이산대수 문제는 특히, 같은 크기의 유한체 K에서의 이산대수 문제보다 훨씬 어렵다고 알려져 있다.

위와 같은 장점을 가진 타원곡선을 본 논문에서 제안한 회로의 일반적 형식은

$$Z = A \cdot B = \sum_{i=0}^{m-1} b_i (A\alpha_i)$$

$$= [\cdots[[b_0(A) + b_1(A\alpha)] + b_2(A\alpha^2)] + \cdots] + b_{m-1(A}\alpha^{m-1}) \quad (1)$$

이다. α에 관한 식을 정리하면

$\alpha_m = p_0 + p_1\alpha + \cdots + p_{m-1}\alpha^{m-1}$ 이므로

$$A\alpha = a_0\alpha + a_1\alpha^2 + \cdots + a_{m-1}\alpha^{m-1}$$

$$= a_0\alpha + a_1\alpha^2 + \cdots + a_{m-1}(p_0\alpha + p_1\alpha_2 + \cdots + p_{m-1}\alpha^{m-1})$$

$$= a_{m-1}p_0 + (a_0 + a_{m-1}p_1)\alpha + (a_1 + a_{m-1}p_2)\alpha^2 + \cdots + (a_{m-2} + a_{m-1}p_{m-1})\alpha^{m-1} \quad (2)$$

(2)식과 함께 (1) 식을 회로로 구현하면 그림 2와 같은 표준형 회로를 얻을 수 있다.[4][5]

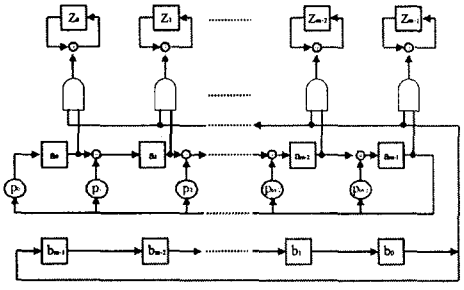

그림 2 순차회로를 이용한 승산기 I

그림 2에서 A와 Z는 모든 bit들이 병렬적으로 처리가 되므로 키 길이와 상관없이 $A \cdot B_i$ mod P는 한 싸이클에 수행이 된다. 그러므로 순차적으로 곱해지는 B의 bit들에 의해 전체 처리시간이 계산되어진다.

p는 field generator의 차수와 같다. 예를 들어 $GF(2^{193})$상에서 $p(x)=x^{193}+x^{15}+1$은 p_{15}, p_0만 연결하면 된다.

또한 위 식을 변형한 다른 승산기를 얻을 수 있다. 식을 보면

$$Z = A \cdot B = \sum_{i=0}^{m-1} b_i (A\alpha_i)$$

$$= (\cdots((b_{m-1}A)\alpha + b_{m-2}A)\alpha + \cdots)\alpha + b_0A$$

이고 그림으로 나타내면 그림 3과 같은 형태이다.

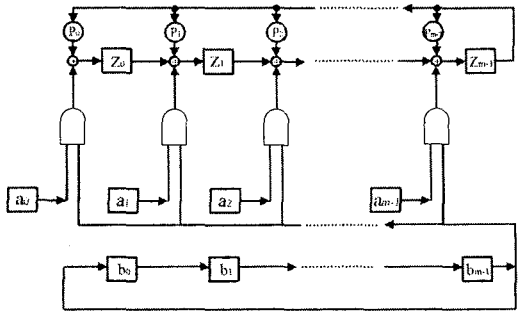

그림 3 순차회로를 이용한 승산기 II

그림 2는 B의 하위 bit부터 계산이 되어지는 반면 그림 3은 상위 bit부터 계산이 되어진다. 이상의 순차 논리회로를 이용한 승산기는 회로가 비교적 간단하다.

III. 제한된 면적과 수행시간의 Tread Off된 회로 제안

3.1 제안된 32bit 모듈화된 회로구조

Bit-Parallel에 바탕을 둔 표준형 승산기는 수행시간이 m clocks인 장점이 있지만 회로의 규모가 3·m 레지스터, m XORs, m ANDs 가 되기 때문에 스마트 카드와 같이 제한된 면적에서는 좀 더 최소화된 구조가 필요하다. 본 논문에서는 그림 4에서와 같이 32bit digit-serial 구조로 제안한다.

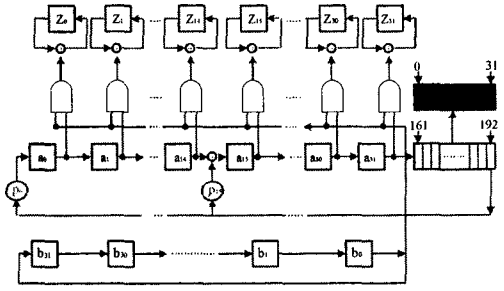

그림 4 승산기 I의 32bit 모듈화된 회로

기능은 승산기 I의 모습을 변형한 것이다. (이 회로는 193bit ECC를 예로 들었다). 회로의 구성은 modular P(x)에 의하여 변형된 a 값이 shift되어 저장되는 레지스터와 a의 32bit, b의 32bit, z의 32bit 레지스터로 구성된다. 또한 메모리를 사용하는데 193bit의 원래의 데이터가 들어있는 메모리와 변형된 데이터가 들어가는 메모리, 그리고 최종값을 저장할 z인 193bit를 필요로 한다. 또한 추가적으로 최종 한 싸이클을 줄이기 위해 a, b, z의 한 bit들을 갖는다. 이 bit들은 최종적으로

192bit를 끝낸 상태에서 193bit를 다시 load 해서 같은 수행을 b의 수만큼, 반복하는 횟수를 줄이고자 마지막 160~193bit는 동시에 수행되도록 설계되었다.

연산 형태를 설명하면, 그림 4를 보면 우측 32bits 레지스터가 두 개 있다. 두 레지스터는 동일하며 하단 레지스터는 처음 32bits data를 load한 상태를 표시하며 상단 레지스터는 32bits shift 연산이 끝난 뒤의 레지스터 상태를 나타낸다. 이 레지스터는 새로운 레지스터를 load 하기 전 변형된 데이터를 저장하는 메모리로 write된다. 다음 load되는 데이터는 하단 레지스터에 load되었던 다음 하위 레지스터를 기준으로 32bits가 load된다. 제일 처음 load되었던 a의 마지막 bit(그림에서는 a_{32})가 연산에 사용이 되면(이때 B는 193bits를 모두 읽어들인 것이다. 마지막 bit는 192가 된다) z의 32bits의 계산은 끝이 나고 z의 마지막 bit (그림에서는 z_{31}) 다음의 bit부터 32bits가 계산되어진다. 여기서 z는 초기에 0으로 초기화를 한다. 이후 원래의 a 값을 load한다.(이때 z의 bit자리와 동일한 데이터를 읽어들인다. 예로 z_{32}에서 z_{63}까지 계산을 하기 위해 초기 a도 a_{32}에서 a_{63}까지 읽어들이게 된다) 이후 우측 레지스터는 그림과 동일한 순서로 load한다. 여기서 우측 레지스터 데이터는 전에 변형된 데이터가 load된다. 이때 B는 마찬가지로 b_0부터 연산에 사용된다.

3.2 보다 감소된 싸이클을 위한 알고리즘

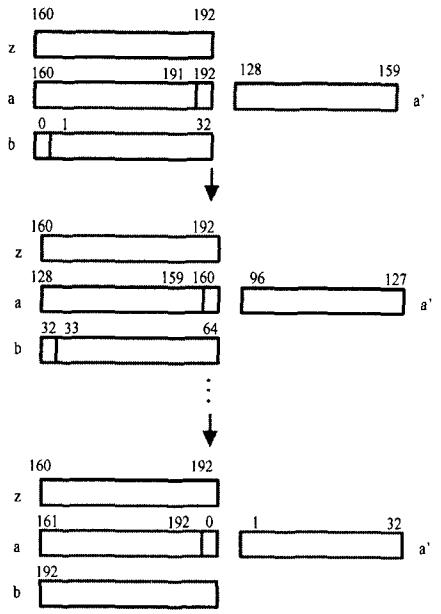

그림 5 최종 1bit cycle 수를 줄이는 알고리즘

그림 5 에서 나타낸 것처럼 마지막 그림의 전 단계에 b의 191bit까지 32bit shift를 끝내고 다시 6번의 연산을 거치지 않고 각 z, a, b를 덧붙여 z192를 같이 계산하기 때문에 최종 단계를 거치지 않는다.

그림 5와 같은 32bit 모듈화 된 승산기의 모듈은 메모리를 추가적으로 요구하지만 기본 193bit 회로를 1/6 수준으로 줄임으로 전체적인 면적은 감소한다. 기존의 160bit를 능가하는 193bit이상의 회로에서 이 시스템의 성능은 증가할 것이다. 이 회로는 CP(critical path)가 무척 짧기 때문에 클럭주파수를 높임으로서 시간도 단축할 수 있는 장점이 있다. 기존 회로 I의 형태를 따서 a는 병렬적으로 b는 순차적으로 입력이 된다.

Ⅳ. 측정결과

제안된 회로와 기존의 Bit-Parallel 회로는 VHDL 코딩과 SYNOPSYS 합성을 통하여 구현되었다. 아래 비교표는 193bit를 기준으로 Bit-Parallel 회로와 제안된 Digit-Serial 회로를 비교했다.

표 2 VHDL과 SYNOPSYS로 구현된 결과

형태	게이트수	cycle 수
그림1 승산기	약 7000	210 cycle,6.93us@33MHz
그림3 승산기	약 1500	1200 cycle,39.6us@33MHz

Ⅳ. 결 론

본 논문에서는 같은 키 길이에서 안전도가 높은 ECC 알고리즘을 소개하였고 이 알고리즘을 이용한 스마트 카드와 같은 휴대용 시스템에서의 제한된 면적 및 처리 시간을 최적화 한 회로 구현을 제안하였다. 제안된 회로는 digit-serial/bit-parallel로 순수한 bit-parallel에 비하여 75%면적 감소를 얻을 수 있다.

참고문헌

[1] J. Guajardo and C. Paar, " Efficient Algorithms for Elliptic Curve Cryptosystems", Advances in Cryptology - CRYPTO 97, B.S Kaliski, ed., pp. 342-356, 1997

[2] B, Sunar and C.K. Koc, "Mastrovito Multiplier for All Trinomials", IEEE Tran. Computers, Vol 48, NO5, pp. 522-527, May,1999

[3] David Naccache David M'Raïhi, "Cryptographic Smart Cards", IEEE MICRO, Vol 16, No 3, pp. 14-23, June, 1996

[4] Edoardo D. Mastrovito, "VLSI Architectures for Computations in Galois Fields", Linköping Studies in Science and Technologz. Dissertations, No 242, pp.35-54, 1991

[5] 이만영, "BCH부호와 Reed-Solomon 부호", 민음사, pp. 21-54

[6] 이인수, "타원곡선 암호시스템에 관한 연구", 연세대학교 대학원 석사 논문, pp3-28, 12월, 1996

[7] 전자통신연구원, " 암호학의 기초" 경문사, pp25-28, 3월, 1999